# KJ11

STACK LIMIT TEST

MD-11-DCKBF-C

EP-DCKBF-C-DL-B



FICHE 1 OF 1

APR 1977

MADE IN USA

# B01

| | | | | | | |
|---|---|---|---|---|---|---|
| EOF1DBKTFDSEQ | 00010000 | 770323 | PDP10 411 | 67HDR1DCKBFCSEQ | 00010000 | 770323 |

```
.REM %
```

IDENTIFICATION

| | |
|---|---|
| PRODUCT CODE: | MAINDEC-11-DCKBF-C-D |
| PRODUCT NAME: | STACK LIMIT TEST<br>STANDARD ON 11/45<br>KJ11-A OPTION ON 11/40 |
| DATE RELEASED: | MARCH, 1977 |
| MAINTAINER: | DIAGNOSTIC GROUP |
| AUTHORS: | ;JOHN ADAMS |

1.0 ABSTRACT

THIS PROGRAM INCREMENTLY TESTS THE STACK LIMIT FUNCTION.

THE PROGRAM USES THE CONTENTS OF LOCATION 176 AS RHE VALUE
OF THE SWITCHES IF NO HARDWARE SWITCH REGISTER IS FOUND.
THE OPERATOR IS RESPONSIBLE FOR LOADING THE DESIRED
VALUE BEFORE STARTING THE PROGRAM. LOCATION 174 WILL
BE USED AS THE SOFTWARE DISPLAY REGISTER.

2.0 REQUIREMENTS

2.1 EQUIPMENT

BASIC 11/45 SYSTEM OR
11/40 WITH STACK LIMIT OPTION

2.2 STORAGE

THIS PROGRAM USES 0 THRU 17500

2.3 PRELIMINARY PROGRAMS

DOAA THRU DOMA

3.0 LOADING PROCEDURE

LOAD PROGRAM USING ABS LOADER.

4.0 STARTING PROCEDURE

LOAD ADDRESS 200, PRESS START, THE PROGRAM WILL LOOP AND RING
BELL AND PRINT '$' ON PASS COMPLETION.

5.0 OPERATING PROCEDURE

5.1 SWITCH SETTINGS

NONE

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

SCOPE IS A MOVE PC,R1 AND STORES THE PC+2 IN R1.

5.2.2 HLT

HLT IS A HALT INSTRUCTION.

6.0 ERRORS

ALL ERRORS WILL CAUSE A HALT.  TRAP AND INTERRUPT ERRORS WILL
CAUSE A HALT AT VECTOR+2.

6.1 ERROR RECOVERY

PRESS CONTINUE TO PROCEDE TO NEXT TEST.

6.2 ERROR LOOPING

TO LOOP ON AN ERROR, PLACE A BRANCH TO THE PREVIOUS SCOPE INSTRUCTION IN PLACE OF THE HALT INSTRUCTION. NOTE THAT IF THE ERROR IS INTERMITTANT THAT THE TEST WILL DROP THRU THE HALT AND PROCEED TO THE NEXT TEST. THEREFORE, TO LOOP THE TEST CONTINUOUSLY REPLACE THE BEQ .+4 INSTRUCTION IMMEDIATELY PRECEEDING THE HALT WITH A BRANCH BACK TO THE PREVIOUS SCOPE.

TO LOOP ON TRAP FAILURES, PATCH IN THE FOLLOWING ROUTINE AT THE ADDRESS OF THE TRAP VECTOR.

```
TRAPVEC:        TRAPVEC+4
TRAPVEC+2:      0
TRAPVEC+4:      012716          ;MOVE SCOPE ADDRESS TO STACK
TRAPVEC+6:      ADDRESS         ;ADDRESS OF PREVIOUS SCOPE
TRAPVEC+10:     000006          ;RETURN TO TEST AT SCOPE
```

RESTORE ALL LOCATIONS BEFORE PROCEEDING TO NEXT TEST.

7.0 RESTRICTIONS

NONE

8.0 MISCELLANEOUS

ON TRAP ERRORS THE STACK POINTER (R6) WILL CONTAIN ADDRESS WHERE THE TRAP OCCURRED.

8.1 EXECUTION TIME

EACH PROGRAM TAKES ABOUT 1 MINUTE.

9.0 PROGRAM DESCRIPTIONS

THIS IS A TEST OF THE STACK LIMIT REGISTER AND INSURES CORRECT OPERATION OF THE RED AND YELLOW ZONE BOUNDARIES. OVERFLOW TRAPS ARE TESTED FOR ALL VALUES OF THE STACK LIMIT REGISTER.

%

```
                                ;TEST DCKBFC- STACK LIMIT REGISTER TEST
                                ;THE STACK LIMIT REGISTER ALLOWS THE 'OVERFLOW' BOUNDARIES TO BE CHANGED.
                                ;FOR EXAMPLE IF THE STACK LIMIT REGISTER IS CLEAR THE BOUNDARY IS AT
                                ;400 (YELLOW ZONE) AND 340 (RED ZONE).  IN ALL CASES THE YELLOW ZONE
                                ;BOUNDARY IS AT 400(8) PLUS THE VALUE IN THE STACK LIMIT REGISTER, AND
                                ;THE RED ZONE BEGINS 20(8) WORDS BELOW THE YELLOW ZONE.  THIS TEST
                                ;CHECKS THAT THE STACK LIMIT IS 400 GREATER THAN THE CONTENTS OF THE
                                ;STACK LIMIT REGISTER (CORE PERMITTING), AND CHECKS THE LENGTH OF THE
                                ;YELLOW ZONE AND THE BEGINNING OF THE RED ZONE.

                                ;STARTING PROCEDURE
                                ;       LOAD ADDRESS=200
                                ;       PRESS START
                                ;       BELL WILL RING WHEN TEST IS COMPLETE

                                ;EQUATE STATEMENTS
000000                          R0=%0
000001                          R1=%1
000002                          R2=%2
000003                          R3=%3
000004                          R4=%4
000005                          R5=%5
000006                          SP=%6
000007                          PC=%7

                                ;REGISTER ADDRESSES
177776                                  PSW=177776              ;ADDRESS OF PROCESSER STATUS WORD
177770                                  UBREAK=177770           ;ADDRESS OF PDP11/45 MICRO BREAK REGISTER
177570                                  DSWR=177570             ;ADDRESS CONSOLE SWITCH REGISTER
177564                                  TPS=177564
177566                                  TPB=177566
177570                                  DDISP=177570            ;ADDRESS OF CONSOLE DISPLAY REGISTER
000000                          HLT=HALT
022626                          POP2=22626
010701                          SCOPE=010701                    ;MOVE PC TO R1
000340                          PRTY7=340                       ;PRIORITY LEVEL 7
000004                          ERRVEC=4                        ;ADDRESS OF ERROR VECTOR

000000                                  .=0

000046                                  .=46
000046  002230                          $ENDAD
000052                                  .=52
000052  000000                          000000

000174                                  .=174
000174  000000                  DISPREG:0
000176  000000                  SWREG:  0

000200                                  .=200
000200  000167  000604                  JMP     START

000500                                  .=500
                                ;TAGS
000500  000000                  ICNT:   0                       ;CONTAINS PASS COUNT
```

```
                                ICNTA:  0
                                SWR:    DSWR
                                DISPLAY:DDISP
                                SPBOT:  760
                                SLR:    177774                  ;ADDRESS OF STACK LIMIT REGISTER
                                SLH:    177775                  ;HIGH (ODD BYTE)
                                TEMP:   0
                                FTITLE: 0                       ;TITLE PRINTED = 1

                        001010                  .=1010
001010  016706  177474          START:  MOV     SPBOT,%6        ;INITIALIZE STACK POINTER
001014  012737  000340  177776          MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
001022  023737  000042  000046          CMP     @#42,@#46       ;ARE WE IN ACT11 AUTOMATIC MODE?
001030  001500                          BEQ     STARTB          ;YES, SKIP THE TITLE
001032  005767  177462                  TST     FTITLE          ;TITLE PRINTED YET?
001036  001037                          BNE     STARTA          ;YES, SKIP TITLE
001040  012700  001064                  MOV     #TITLE,R0       ;GET MESSAGE ADDRESS
001044  105710                  1$:     TSTB    (0)             ;END OF MESSAGE?
001046  001433                          BEQ     STARTA          ;YES; GET OVER THE ASCII
001050  105767  176510                  TSTB    TPS
001054  100375                          BPL     .-4
001056  112067  176504                  MOVB    (0)+,TPB        ;PRINT CHARACTER
001062  000770                          BR      1$
001064  005015  042177  045503  TITLE:  .ASCIZ  <15><12><177>/DCKBF-C, STACK LIMIT REGISTER TEST/<15><12><177>
001072  043102  041455  020054
001100  052123  041501  020113
001106  044514  044515  020124
001114  042522  044507  052123
001122  051105  052040  051505
001130  006524  077412     000
                        001136                  .EVEN
001136  012767  000001  177354  STARTA: MOV     #1,FTITLE       ;SET TITLE PRINTED FLAG
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
001144  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
001150  012737  001204  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
001156  012767  177570  177320          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
001164  012767  177570  177314          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
001172  022777  177777  177304          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
001200  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
001202  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
001204  012716  001212          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
001210  000002                          RTI
001212  012767  000176  177264  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
001220  012767  000174  177260          MOV     #DISPREG,DISPLAY
001226  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
001232  005067  177242          STARTB: CLR     ICNT            ;CLEAR PASS COUNT
001236  005767  177240                  TST     ICNTA
001242  001412                          BEQ     BEGIN
001244  022767  177777  000762          CMP     #-1,$ENDAD+4
001252  001006                          BNE     BEGIN
001254  022767  001477  177216          CMP     #1477,ICNT
001262  001002                          BNE     BEGIN
001264  005067  177212                  CLR     ICNTA
```

```
                                BEGIN:  MOV     SPBOT,%6        ;INITIALIZE STACK POINTER
                                        MOV     ICNT,@DISPLAY   ;DISPLAY PASS COUNT
                                        BIT     #400,@SWR       ;LOAD PDP11/45 MICRO BREAK REGISTER?
                                        BEQ     .+10
                                        MOVB    @SWR,@#UBREAK   ;LOAD MICRO BREAK REG WITH SPO-7

                                ;CHECK THAT CP CAN TIME OUT TRAP
                                        MOV     #TORET,@#ERRVEC ;LOAD TIMEOUT TRAP VECTOR
                                        CLR     @#173000        ;ADDRESS 173000 ALWAYS TIMES OUT ON
                                                                ;DATIP/DATO BUS CYCLE
                                        HLT                     ;ERROR! FAILED TO TIME OUT TRAP
                                        BR      BEGIN           ;LOOP TEST
                                TORET:  CMP     (6)+,(6)+       ;RESTORE THE STACK

                                ;TEST THAT THE STACK LIMIT REGISTER CAN BE REFERENCED USING DATI,
                                ;DATIP/DATO
                                T0:     SCOPE
                                        MOV     #T0A,ERRVEC     ;LOAD ERROR VECTOR
                                        CLR     ERRVEC+2
                                        MOV     @SLR,@#177774   ;REFERENCE STACK LIMIT REGISTER
                                        BR      T0B             ;GO TO NEXT TEST
                                T0A:    POP2
                                        HLT                     ;ERROR:  CANNOT REFERENCE STACK LIMIT REG.
                                        BR      T0              ;LOOP TEST IF ERROR

                                ;USING DATI, DATIP/DATOB

                                T0B:    MOV     #T0C,ERRVEC     ;LOAD ERROR VECTOR
                                        MOVB    @SLR,@SLH       ;REFERENCE ODD BYTE
                                        BR      T0D
                                T0C:    POP2
                                        HLT                     ;ERROR! CANNOT REFERENCE STACK LIMIT
                                        BR      T0B             ;USING BYTE INSTRUCTION.

                                T0D:    MOV     #6,ERRVEC       ;RESTORE ERROR TRAP VECTOR
                                        BR      T1              ;GO TO NEXT TEST

                                ;TEST THAT EACH BIT OF THE STACK LIMIT REGISTER BITS CAN BE SET
                                ;AND CLEARED. THIS TEST ROTATES A BIT THROUGH THE STACK LIMIT REGISTER.

                                T1:     SCOPE
                                        MOV     #400,R2         ;LOAD TEST VALUE
                                T1A:    MOV     R2,@SLR         ;LOAD TEST VALUE INTO STACK LIM. REG.
                                        MOV     @SLR,R0         ;GET RESULT
                                        CMP     R0,R2           ;CHECK RESULT
                                        BEQ     .+4             ;BRANCH IF RESULT IS CORRECT
                                        HLT                     ;ERROR! INCORRECT RESULT. R2 HAS CORRECT
                                                                ;RESULT AND R0 HAS INCORRECT RESULT.
                                        CLR     @SLR            ;CLEAR STACK LIM. REG.
                                        MOV     @SLR,R0         ;GET AND CHECK RESULT
                                        BEQ     .+4
                                        HLT                     ;ERROR! INCORRECT RESULT
                                        ASL     R2              ;SHIFT TEST VALUE
                                        BCC     T1A             ;BRANCH IF NOT DONE
                                        BR      T2
                                        .=1410
```

```
                                ;THIS TEST INCREMENTS THE STACK LIMIT REGISTER

001410  010701          T2:     SCOPE
001412  005067  177100          CLR     TEMP
001416  005077  177070          CLR     @SLR            ;CLEAR STACK LIMIT REGISTER
001422  017700  177064  T2A:    MOV     @SLR,R0         ;GET RESULT
001426  020067  177064          CMP     R0,TEMP         ;CHECK RESULT
001432  001401                  BEQ     .+4
001434  000000                  HLT                     ;ERROR! STACK LIM. REG. WAS INCORRECT DATA
                                                        ;TEMP # CORRECT RESULT
001436  105277  177052          INCB    @SLH            ;INCREMENT VALUE IN STACK LIM. REG.
001442  105267  177051          INCB    TEMP+1          ;INCREMENT TEST VALUE
001446  001365                  BNE     T2A             ;BRANCH IF ALL VALUES NOT TESTED

                                ;TEST THAT RESET CLEARS THE STACK LIMIT REGISTER

001450  010701          T3:     SCOPE
001452  012777  177777  177032  MOV     #-1,@SLR        ;PRESET SLR
001460  017700  177026          MOV     @SLR,R0         ;SAVE SLR
001464  022700  177400          CMP     #177400,R0      ;CHECK THAT SLR WAS LOADED
001470  001401                  BEQ     .+4
001472  000000                  HLT                     ;! SLR FAILED TO LOAD
001474  000005                  RESET                   ;RESET CLEARS SLR
001476  017700  177010          MOV     @SLR,R0         ;GET RESULT OF RESET AND CHECK RESULT
001502  001402                  BEQ     T4              ;GO TO NEXT TEST IF RESET CLEARED SLR
001504  000000                  HLT                     ;ERROR! RESET FAILED TO CLEAR SLR
001506  000760                  BR      T3              ;LOOP TEST IF ERROR

                                ;TEST THAT THE CLEAR INSTRUCTION CLEARS THE STACK LIMIT REGISTER

001510  010701          T4:     SCOPE
001512  112777  177777  176774  MOVB    #-1,@SLH        ;PRESET ODD BYTE
001520  017700  176766          MOV     @SLR,R0         ;GET RESULT
001524  022700  177400          CMP     #177400,R0      ;CHECK RESULT
001530  001401                  BEQ     .+4
001532  000000                  HLT                     ;ERROR! SLR DID NOT PRESET
001534  005077  176752          CLR     @SLR
001540  017700  176746          MOV     @SLR,R0         ;GET RESULT OF CLEAR & BRANCH IF CLEAR
001544  001402                  BEQ     T5              ;GO TO NEXT TEST
001546  000000                  HLT                     ;ERROR! CLR INST FAILED TO CLEAR SLR
001550  000757                  BR      T4              ;LOOP TEST IF ERROR


                                ;TEST THAT AN OVERFLOW ERROR OCCURS FOR ALL STACK LIMIT REGISTER VALUES.
                                ;(PROVIDED CORE IS AVAILABLE).

001552  010701          T5:     SCOPE
001554  012702  000010          MOV     #10,R2          ;INITIALIZE STACK VALUE AND
001560  012703  177400          MOV     #-400,R3        ;STACK LIMIT REGISTER VALUE
001564  062702  000400  T5A:    ADD     #400,R2         ;LOAD NEW STACK VALUE
001570  062703  000400          ADD     #400,R3         ;AND NEW STACK LIM. REG. VALUE
001574  005037  000000          CLR     @#0             ;CLEAR ADDRESS 0
001600  016706  176704          MOV     SPBOT,SP        ;INITIALIZE THE STACK POINTER
001604  012767  002072  176172  T5B:    MOV     #LIMX,ERRVEC    ;LOAD TIME OUT TRAP
001612  016204  177776          MOV     -2(2),R4        ;SAVE STACK LOCATIONS
```

```
   375  001616  016205  177774                 MOV     -4(2),R5        ;EXIT TEST IF EITHER TEST TIMES OUT
   376  001622  012767  001646  176154         MOV     #LIMA,ERRVEC    ;LOAD OVERFLOW VECTOR
   377  001630  010206                         MOV     R2,SP           ;LOAD STACK POINTER
   378  001632  010377  176654                 MOV     R3,@SLR         ;LOAD STACK LIM. REG.
   379  001636  016666  177770  177770         MOV     -10(6),-10(6)   ;REFERENCE LIMIT ADDRESS
   380  001644  000401                         BR      .+4             ;SHOULD NOT HAVE TRAPPED
   381  001646  000000                 LIMA:   HLT                     ;ERROR! REFERENCE TO LIMIT ADDRESS
   382                                                                 ;CAUSED AN OVERFLOW
   383  001650  010206                         MOV     R2,SP           ;REPOINT STACK POINTER
   384  001652  012767  001670  176124         MOV     #LIMB,ERRVEC    ;REPOINT OVERFLOW VECTOR
   385  001660  016666  177766  177766         MOV     -12(6),-12(6)   ;REFERENCE FIRST 'YELLOW' ADDRESS
   386  001666  000000                         HLT                     ;ERROR! SHOULD HAVE TRAPPED
   387  001670  005737  000000         LIMB:   TST     @#0             ;WAS IT 'RED' OVERFLOW?
   388  001674  001401                         BEQ     .+4
   389  001676  000000                         HLT                     ;ERROR! A 'RED' OVERFLOW OCCURRED WHEN
   390                                                                 ;A 'YELLOW' ADDRESS WAS REFERENCED.
   391  001700  010206                         MOV     R2,SP           ;REPOINT STACK POINTER
   392  001702  005037  000000                 CLR     @#0             ;CLEAR ADDRESS 0
   393  001706  012767  001724  176070         MOV     #LIMC,ERRVEC    ;REPOINT OVERFLOW VECTOR
   394  001714  016666  177730  177730         MOV     -50(6),-50(6)   ;REFERENCE LAST 'YELLOW' ADDRESS
   395  001722  000000                         HLT                     ;ERROR! SHOULD HAVE TRAPPED
   396  001724  005737  000000         LIMC:   TST     @#0             ;WAS IT 'RED' OVERFLOW
   397  001730  001401                         BEQ     .+4
   398  001732  000000                         HLT                     ;ERROR! A 'RED' OVERFLOW OCCURRED WHEN
   399                                                                 ;THE LAST 'YELLOW' ADDRESS WAS REFERENCED
   400  001734  005037  000000                 CLR     @#0             ;CLEAR ADDRESS 0
   401  001740  010206                         MOV     R2,SP           ;REPOINT THE STACK POINTER
   402  001742  012767  002010  176034         MOV     #LIMD,ERRVEC    ;REPOINT THE OVERFLOW TRAP
   403  001750  016267  177726  176540         MOV     -52(2),TEMP     ;GET 'RED' LOCATION
   404  001756  005166  177726                 COM     -52(6)          ;REFERENCE 'RED' ADDRESS
   405  001762  000000                 LIMCC:  HLT                     ;ERROR! NO OVERFLOW TRAP WHEN 'RED' ADDRESS
   406  001764  000411                         BR      LIMD                    ;WAS REFERENCED
   407          002010                         .=2010
   408  002010  026267  177726  176500 LIMD:   CMP     -52(2),TEMP     ;WAS INSTRUCTION ABORTED?
   409  002016  001401                         BEQ     .+4
   410  002020  000000                         HLT                     ;ERROR! COM -52(6) WAS ALLOWED TO CHANGE
   411                                                                 ;A 'RED' ADDRESS.
   412  002022  016762  176470  177726         MOV     TEMP,-52(2)     ;RESTORE 'RED' LOCATION IN ANY EVENT
   413  002030  005706                         TST     %6              ;WAS STACK POINTER CHANGED TO 0?
   414  002032  001401                         BEQ     .+4
   415  002034  000000                         HLT                     ;'RED' OVERFLOW DID NOT ASSUME NEW STACK
   416  002036  022737  001762  000000         CMP     #LIMCC,@#0      ;IS RETURN ADDRESS ON NEW STACK?
   417  002044  001401                         BEQ     .+4
   418  002046  000000                         HLT                     ;ERROR! RETURN ADDRESS NOT SAVED ON NEW
   419                                                                 ;STACK
   420  002050  005077  176436                 CLR     @SLR            ;GET READY TO GET NEW VALUES FOR TEST
   421  002054  016706  176430                 MOV     SPBOT,SP        ;INITIALIZE THE STACK POINTER
   422  002060  010462  177776                 MOV     R4,-2(2)        ;RESTORE STACK ADDRESS DATA
   423  002064  010562  177774                 MOV     R5,-4(2)
   424  002070  000635                         BR      T5A             ;GET NEW VALUES
   425  002072  012767  000006  175704 LIMX:   MOV     #6,ERRVEC       ;RESTORE TIME OUT TRAP VECTOR
   426
   427
   428  002100  005767  176376                 TST     ICNTA                   ;TEST FOR
   429  002104  001013                         BNE     END
   430  002106  022767  177777  000120         CMP     #-1,$ENDAD+4            ;SCRIPT
```

```
   431  002114  001417                          BEQ     DONE                    ;CONDITION
   432  002116  026767  175724  175716          CMP     46,42                   ;AND
   433  002124  001003                          BNE     END                     ;SET
   434  002126  005267  176350                  INC     ICNTA                   ;FLAG
   435  002132  000410                          BR      DONE
   436  002134  005267  176340          END:    INC     ICNT
   437  002140  026727  176334  001500          CMP     ICNT,#1500      ;HAVE 1500 PASSES BEEN COMPLETED
   438  002146  001402                          BEQ     DONE
   439  002150  000167  177114                  JMP     BEGIN
   440  002154  012767  000007  175404  DONE:   MOV     #7,TPB          ;RING BELL
   441  002162  105767  175376                  TSTB    TPS
   442  002166  100375                          BPL     .-4
   443  002170  012767  000052  175370          MOV     #52,TPB         ;PRINT '*'
   444  002176  105767  175362                  TSTB    TPS
   445  002202  100375                          BPL     .-4
   446  002204  012767  000000  175354          MOV     #0,TPB          ;INSERT
   447  002212  105767  175346                  TSTB    TPS             ;NULL
   448  002216  100375                          BPL     .-4             ;CHARACTER
   449  002220  013702  000042                  MOV     @#42,%2         ;GET MONITOR RETURN ADDRESS
   450  002224  001410                          BEQ     DONE1           ;DO NOT RETURN IF (42)=0
   451  002226  000005                          RESET
   452  002230  004712                  $ENDAD: JSR     7,(2)           ;RETURN TO MONITOR
   453  002232  000240                          NOP                     ;ACT11
   454  002234  000240                          NOP                     ;OVERLAY
   455  002236  000240                          NOP                     ;AREA
   456  002240  005000                          CLR     R0              ;DELAY FOR ACT11
   457  002242  005200                          INC     R0
   458  002244  001376                          BNE     .-2
   459  002246  000167  176536          DONE1:  JMP     START           ;RESTART
   460
   461          000001                          .END
```

```
BEGIN   001270          ICNT    000500          SCOPE = 010701          TORET   001336          T2A     001422
DDISP = 177570          ICNTA   000502          SLH     000514          TPB   = 177566          T3      001450
DISPLA  000506          LIMA    001646          SLR     000512          TPS   = 177564          T4      001510
DISPRE  000174          LIMB    001670          SPBOT   000510          TO      001340          T5      001552
DONE    002154          LIMC    001724          START   001010          TOA     001364          T5A     001564
DONE1   002246          LIMCC   001762          STARTA  001136          TOB     001372          T5B     001604
DSWR  = 177570          LIMD    002010          STARTB  001232          TOC     001410          UBREAK= 177770
END     002134          LIMX    002072          SWR     000504          TOD     001416          $ENDAD  002230
ERRVEC= 000004          POP2  = 022626          SWREG   000176          T1      001426          .     = 002252
FTITLE  000520          PRTY7 = 000340          TEMP    000516          T1A     001434
HLT   = 000000          PSW   = 177776          TITLE   001064          T2      001410

. ABS.  002252      000
```

```
 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DSKZ:DCKBFC,DSKZ:DCKBFC/SOL=DSKZ:DCKBFC.SML,DSKZ:DCKBFC.P11
 RUN-TIME: 20 21 .1 SECONDS
 RUN-TIME RATIO: 78/42=1.8
 CORE USED:  31K  (61 PAGES)
```